Ann Oper Res (2014) 213:203–220
DOI 10.1007/s10479-012-1155-9

Simulated annealing for financing cost distribution based project payment scheduling from a joint perspective

Zhengwen He · Nengmin Wang · Pengxiang Li

Published online: 16 May 2012

Abstract Based on the distribution of the project financing cost over the contractor and the client, this paper involves the project payment scheduling problem from a joint perspective of the two parties. In the problem, the project financing cost is defined as the expense for raising money from the outside or the opportunity cost of the capital devoted into the project and the objective is to find the project payment schedule that can not only maximize the joint revenue of the two parties but also be accepted by them. Based on the characteristics of the problem, an optimization model consisting of two submodels is constructed using the activity-based method. For the strong NP-hardness of the problem, two simulated annealing algorithms with different searching structures are developed and compared with the multi-start iterative improvement method on the basis of a computational experiment performed on a data set generated randomly. The results show that the simulated annealing algorithm with the nested loop module seems to be the most promising algorithm for solving the defined problem especially when the scale of the problem becomes larger. In addition, the influences of some key parameters on the computational results are investigated through the full factorial experiment and a few useful conclusions are drawn.

Keywords Project scheduling · Progress payments · Financing cost distribution · Simulated annealing

1 Introduction

As one of the most important aspects in project management, the financial issue is attracting more and more researchers' attention. Russell (1970) is the first researcher who attempts to

Z. He (✉) · N. Wang · P. Li
School of Management, Xi'an Jiaotong University, Xi'an 710049, China
e-mail: zhengwenhe@mail.xjtu.edu.cn

N. Wang
e-mail: wangnm@mail.xjtu.edu.cn

P. Li
e-mail: lipengx@mail.xjtu.edu.cn

model the financial aspects of project management with the objective of maximizing the net present value (NPV) of cash flows in the project. Since Russell's pioneering work, more and more efforts have been spent on the development of schedules that take financial aspects into account and this has led to a large amount of models and algorithms. For an overview of project scheduling with the objective of the maximization of the NPV in general, we refer the reader to Icmeli et al. (1993), Özdamar and Ulusoy (1995), Herroelen et al. (1997), Brucker et al. (1999), Kolisch and Padman (2001), Demeulemeester and Herroelen (2002), Błażewicz et al. (2007), Drezet (2008), Hartmann and Briskorn (2010), and Węglarz et al. (2011).

The project payment scheduling problem (PPSP), which can be considered as a new branch of the Max-NPV project scheduling problem, involves how to schedule progress payments so as to maximize the NPV of the contractor or/and the client. Considering the single-mode PPSP where activities can only be executed with one mode, Dayanand and Padman (1997) introduce several deterministic models to maximize the contractor's NPV where a deadline is imposed and the number and the total amount of payments from the client are fixed. Owing to the combinatorial nature of the problem, Dayanand and Padman (2001a) present a two-stage procedure for the single-mode PPSP in which a set of payments is determined using a simulated annealing algorithm in the first stage and in the second stage, activities are rescheduled to improve the NPV. Dayanand and Padman (2001b) also establish several mixed integer linear programming models from the client's viewpoint according to practical payment rules and draw the conclusion that the client obtains the greatest benefit by scheduling the project for early completion such that the payments are not made at regular intervals. On the basis of the fact that the timing of payments and the completion times of activities are determined by the client and the contractor respectively, Szmereskovsky (2005) develops a branch-and-bound procedure for a novel single-mode PPSP model where the payment schedule is chosen by the client and the contractor protects his interests by selecting the activity schedule to maximize his own NPV and by rejecting the payment schedule if his NPV does not exceed a minimum amount.

Other researchers devote their attentions into the multi-mode PPSP where activities can be accomplished in more than one way. Ulusoy and Cebeli (2000) propose a double-loop genetic algorithm in which the outer loop represents the client and the inner loop the contractor to find an equitable payment schedule of the project for the two parties of contract. Kavlak et al. (2009) extend the work of Ulusoy and Cebeli (2000) by introducing a bargaining power concept into the objective of the problem and use simulated annealing algorithm and genetic algorithm approaches as solution procedures for the problem. He and Xu (2008) analyze the effects of the bonus-penalty structure on project payment scheduling and find that the structure can enhance the flexibility of project payment scheduling and be helpful for the two parties of contract to get more profits from the project synchronously. He et al. (2009) develop two heuristic algorithms for the multi-mode PPSP, namely simulated annealing and tabu search, and compare their performances on a data set constructed by ProGen project generator.

In practice, the preferences of the contractor and the client on payment scheduling are often contradicted with each other and the coordination of the conflict between them is always a hard work. Therefore, how to schedule payments so as to make both of the two parties be willing to accept is a problem worthwhile to be investigated intensively. Aiming at such a problem, this paper investigates the PPSP based on the distribution of the project financing cost over the contractor and the client from a joint perspective of the two parties since given the contract price of the project, payment scheduling is associated with the distribution of the project financing cost impliedly. In this problem, the project financing cost is defined as the expense for raising money from the outside or the opportunity cost of the capital devoted into the project and its distribution is determined by the relative bargaining power of the contractor and the client in contract negotiation. The activities can be performed with one of several alternative modes and the project must be finished no later than a deadline. At the completion time of the project, there exists a sole cash inflow for the client, namely the client's expected revenue which is generated by discounting the project future incomes back to this time. The contract price of the project and the total number of payments are given in advance and during the execution of the project, the amount of payments is determined by the payment proportion and the contractor's earned value accumulated up to the payment time. On the basis of the condition aforementioned, the task of this research is to find the project payment schedule that can not only maximize the joint revenue of the contractor and the client but also be accepted by both of the two parties.

Based on the above identification, we name the studied problem the financing cost distribution based PPSP (FCDPPSP) and divide it into the following two subproblems.

- Subproblem 1: How to determine the execution modes and the completion times of activities so that the joint revenue of the contractor and the client is maximized.
- Subproblem 2: How to arrange payments so as to distribute the project financing cost between the two parties according to their relative bargaining power in the contract negotiation.

We believe that the FCDPPSP, which has not been investigated so far to the best of our knowledge, may be more realistic than the PPSPs studied before because it takes the preferences of the contractor and the client into account concurrently and can facilitate the two parties to achieve a more profitable outcome.

The remainder of this paper is organized as follows. In the next section, we provide the optimization model of the FCDPPSP and an illustrative example. The simulated annealing algorithms are developed in Sect. 3 and Sect. 4 devotes to the computational experiment. Section 5 concludes the paper.

2 Problem formulation

2.1 Optimization model

Consider a project represented as an activity-on-the-node (AoN) network $G = (N, A)$, where the set of nodes, N, represents activities, and the set of arcs, A, represents finish-start precedence constraints with a time lag of zero. The activities are numbered from the dummy start activity 1 to the dummy end activity n and each activity i, $i = 1, 2, \dots, n$, has to be executed in one of M_i modes. The duration and the cost of activity i executed in mode m are d_{im} and c_{im} respectively, and we assume that c_{im} occurs at the completion of activity i. The earned value of activity i is v_i, which represents the amount of the compensation that the contractor can obtain from the client for finishing activity i. During the execution of the project, the client makes payments to the contractor at the completion of some certain activities named payment activities. The number of payments is K $(K \leq n)$ and the amount of the k-th ($k = 1, 2, \dots, K - 1$) payment, p_k, equals the product of the contractor's earned value accumulated by the payment time and the compensation proportion θ $(0 \leq \theta \leq 1)$. The last payment K must be arranged at the completion of the dummy end activity n, and its amount is determined by the formula: $p_K = U - \sum_{k=1}^{K-1} p_k$ where U $(U = \sum_{i=1}^{n} v_i)$ is the contract price of the project. The client's expected revenue, which is assumed to occur at the completion of the project, is ER. The project deadline and the interest rate per period are denoted as D and α, respectively.

The three groups of decision variables in the FCDPPSP are defined below,

$$x_{im}=\begin{cases}1 & \text{if activity } i \text{ is performed with mode } m\\ 0 & \text{otherwise}\end{cases}\quad i=1,2,\dots,n;\ m=1,2,\dots,M_i$$

$$y_{it}=\begin{cases}1 & \text{if activity } i \text{ is finished at period } t\\ 0 & \text{otherwise}\end{cases}\quad i=1,2,\dots,n;\ t=1,2,\dots,D$$

$$z_{ki}=\begin{cases}1 & \text{if the } k\text{-th payment is attached to activity } i\\ 0 & \text{otherwise}\end{cases}\quad k=1,2,\dots,K;\ i=1,2,\dots,n.$$

Based on the above definition of decision variables, we define the three decision vectors further for the sake of description:

$$\Omega=(m:x_{im}=1,i=1,2,\dots,n),\qquad \Gamma=(t:y_{it}=1,i=1,2,\dots,n),\quad \text{and}$$
$$\Psi=(z_{ki},i=1,2,\dots,n).$$

Now, let β be the financing cost rate per period then the project financing cost, FC, can be calculated by

$$FC=\sum_{i=1}^{n}\left\{\left[\sum_{m=1}^{M_i}(x_{im}c_{im})\right]\exp\left[\beta\left(\sum_{t=EF_n}^{LF_n}(y_{nt}t)-\sum_{t=EF_i}^{LF_i}(y_{it}t)\right)\right]\right\}-TC. \qquad (1)$$

Where, TC ($TC=\sum_{i=1}^{n}\sum_{m=1}^{M_i}(x_{im}c_{im})$) is the contractor's expense for finishing the project and $[EF_i,LF_i]$ is activity i's completion time window calculated using the critical path method (CPM). During the execution of the project, the client makes payments to the contractor and thus bears a part of the project financing cost, FC_{client},

$$FC_{client}=\sum_{k=1}^{K}\left\{p_k\exp\left[\beta\left(\sum_{t=EF_n}^{LF_n}(y_{nt}t)-\sum_{i=1}^{n}\left(z_{ki}\sum_{t=EF_i}^{LF_i}(y_{it}t)\right)\right)\right]\right\}-U. \qquad (2)$$

Hence, the financing cost of the contractor is only FC_{cont}, $FC_{cont}=FC-FC_{client}$. We call λ ($\lambda=\frac{FC_{client}}{FC}$), which reflects the proportion of the project financing cost undertaken by the client in fact, the distribution proportion of the project financing cost over the contractor and the client. The relative bargaining power of the two parties in the contract negotiation is represented as RBP ($0\le RBP\le 1$), which takes 0 if the client holds an overwhelming superiority over the contractor and 1 if the reverse is true.

Based on the above definitions, the optimization model of the FCDPPSP, which consists of two submodels, is constructed as follows.

- Submodel 1

$$\begin{aligned}\text{Maximize } NPV_{joint}&=NPV_{cont}+NPV_{client}\\&=ER\exp\left[-\alpha\sum_{t=EF_n}^{LF_n}(y_{nt}t)\right]\\&\quad-\sum_{i=1}^{n}\left\{\left[\sum_{m=1}^{M_i}(x_{im}c_{im})\right]\exp\left[-\alpha\sum_{t=EF_i}^{LF_i}(y_{it}t)\right]\right\}\end{aligned} \qquad (3)$$

$$\text{Subject to } \sum_{i=1}^{M_i} x_{im} = 1 \quad i = 1, 2, \dots, n \tag{4}$$

$$\sum_{t=EF_i}^{LF_i} y_{it} = 1 \quad i = 1, 2, \dots, n \tag{5}$$

$$\sum_{t=EF_i}^{LF_i} (y_{it}t) \le \sum_{t=EF_j}^{LF_j} (y_{jt}t) - \sum_{m=1}^{M_j} (x_{jm}d_{jm}) \quad \forall (i, j) \in A \tag{6}$$

$$\sum_{t=EF_n}^{LF_n} (y_{nt}t) \le D \tag{7}$$

$$x_{im}, y_{it} \in \{0, 1\}. \tag{8}$$

Where, NPV_{cont} and NPV_{client} are the NPVs of the contractor and the client respectively, NPV_{joint} is the joint revenue of the two parties.

- Submodel 2

$$\text{Minimize } \mathit{Diff} = |\lambda - RBP| \tag{9}$$

$$\text{Subject to } \sum_{i=1}^{n-1} z_{ki} = 1 \quad k = 1, 2, \dots, K-1 \tag{10}$$

$$\sum_{k=1}^{K} z_{ki} \le 1 \quad i = 1, 2, \dots, n \tag{11}$$

$$p_k = \theta \left[\sum_{i=1}^{n} \left(v_i \sum_{t=0}^{T_k} y_{it} \right) - \sum_{i=1}^{n} \left(v_i \sum_{t=0}^{T_{k-1}} y_{it} \right) \right] \quad k = 1, 2, \dots, K-1 \tag{12}$$

$$p_K = U - \sum_{k=1}^{K-1} p_k \tag{13}$$

$$z_{ki} \in \{0, 1\}. \tag{14}$$

Where, T_k ($T_k = \sum_{i=1}^{n}[z_{ki} \sum_{t=EF_i}^{LF_i}(y_{it}t)]$) and T_{k-1} ($T_{k-1} = \sum_{i=1}^{n}[z_{k-1,i} \sum_{t=EF_i}^{LF_i}(y_{it}t)]$) are the occurrence times of the k-th payment and the $(k-1)$-th payment, respectively.

Submodels 1 and 2 correspond to subproblems 1 and 2 respectively and their solutions constitute together a solution for the FCDPPSP. In submodel 1, the decision variables include x_{im} and y_{it} and the objective is to maximize NPV_{joint}, which is equivalent to the present value of ER minus the present values of c_{im}. Constraints (4) select a mode for each activity exactly. Constraints (5) make sure that the completion time of each activity must be within its completion time window while the precedence feasibility is maintained by (6). A deadline is imposed to the project by constraint (7) and constraints (8) define the binary status of the decision variables. Through solving submodel 1, we may obtain the optimal x_{im} and y_{it}, which are the input data for submodel 2 where the decision variables are z_{ki}. In submodel 2, the objective is to minimize the absolute difference between λ and RBP, i.e. Diff, making the distribution proportion of the project financing cost over the contractor and the client match their relative bargaining power as well as possible. Constraints (10) attach payments to activities and (11) secure that at a certain activity, only one payment can be arranged at most. Constraints (12) compute amounts of payments while constraint (13) forces that the

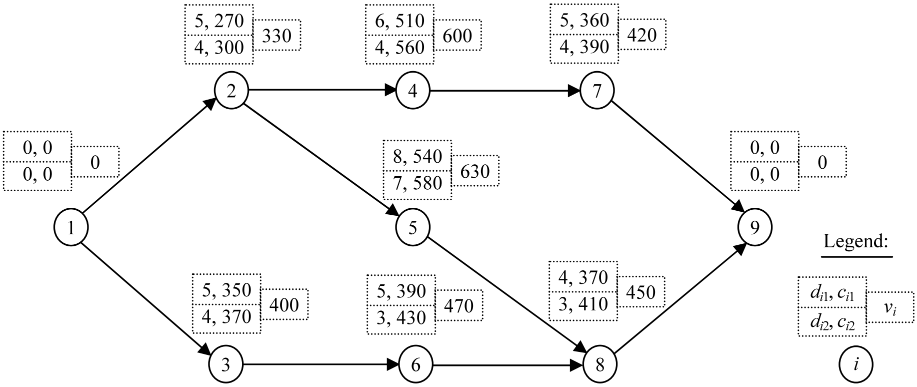

Fig. 1 An example

total amount of payments equals the contract price. Constraints (14) give the definition field of the decision variables.

Without any loss of generality, let $ER = 0$ and $\alpha = 0$, then the objective of submodel 1 can be transformed to the maximization of $-\sum_{i=1}^{n}\sum_{m=1}^{M_i}(x_{im}c_{im})$, i.e. the minimization of TC. This means that when $ER = 0$ and $\alpha = 0$ subproblem 1 can be simplified to assigning the modes and the completion times of activities to minimize the total expense under the constraint of the project deadline. Recalling that P_C|T in DTCTP (De et al. 1995) is to identify a realization of activities' modes that minimizes the cost of the project subject to a given deadline, we can conclude immediately that subproblem 1 is a generalization of P_C|T with the consideration of the client's expected revenue and the time value of money. Because P_C|T has been proven to be strongly NP-hard for general project networks (De et al. 1997), subproblem 1 and thus the FCDPPSP must be strongly NP-hard as well.

2.2 An example

We use an example to illustrate the FCDPPSP identified above. The AoN network of the example is depicted in Fig. 1 where activities 1 and 9 are the dummy start activity and the dummy end activity respectively. The activities in the project can be performed with two modes and their earned values as well as durations and costs under different modes are labeled with nodes. Other data of the project are as follows: $K = 3$, $\theta = 0.8$, $D = 16$, $ER = 6000$, $\alpha = 0.03$, $\beta = 0.04$, $RBP = 0.8$. To demonstrate that a cooperative attitude may allow both the contractor and the client to get more profits from the project, we give the desirable project payment schedules under two different cases and compare the results produced below.

- The cooperative case

In this case, we suppose that the two parties know their relative bargaining power well and take a cooperative attitude in project payment scheduling. It is easy to understand that this case satisfies the assumptions of the model constructed above, and the desirable payment schedule of the project can be worked out easily: $\Omega = (1, 2, 1, 1, 2, 1, 2, 2, 1)$, $\Gamma = (0, 4, 6, 10, 11, 11, 14, 14, 14)$, and $\Psi = (0, 0, 1, 0, 1, 0, 0, 0, 1)$. Under the desirable solution, NPV_{joint}, FC, FC_{client}, and FC_{cont} are calculated below:

$$\begin{aligned} NPV_{joint} &= ER \times \exp(-\alpha \times 14) - \big[c_{11} \times \exp(-\alpha \times 0) + c_{22} \times \exp(-\alpha \times 4) \\ &\quad + c_{31} \times \exp(-\alpha \times 6) + c_{41} \times \exp(-\alpha \times 10) + c_{52} \times \exp(-\alpha \times 11) \\ &\quad + c_{61} \times \exp(-\alpha \times 11) + c_{72} \times \exp(-\alpha \times 14) + c_{82} \times \exp(-\alpha \times 14) \\ &\quad + c_{91} \times \exp(-\alpha \times 14)\big] = 1783.05, \\ FC &= \big\{c_{11} \times \exp\big[\beta \times (14\text{–}0)\big] + c_{22} \times \exp\big[\beta \times (14\text{–}4)\big] + c_{31} \times \exp\big[\beta \times (14\text{–}6)\big] \\ &\quad + c_{41} \times \exp\big[\beta \times (14\text{–}10)\big] + c_{52} \times \exp\big[\beta \times (14\text{–}11)\big] + c_{61} \times \exp\big[\beta \times (14\text{–}11)\big] \\ &\quad + c_{72} \times \exp\big[\beta \times (14\text{–}14)\big] + c_{82} \times \exp\big[\beta \times (14\text{–}14)\big] \\ &\quad + c_{91} \times \exp\big[\beta \times (14\text{–}14)\big]\big\} - TC = 491.70, \\ FC_{client} &= \big\{p_1 \times \exp\big[\beta \times (14\text{–}6)\big] + p_2 \times \exp\big[\beta \times (14\text{–}11)\big] \\ &\quad + p_3 \times \exp\big[\beta \times (14\text{–}14)\big]\big\} - U = 393.64, \quad \text{and} \\ FC_{cont} &= FC - FC_{client} = 98.06. \end{aligned}$$

Where, $TC = c_{11} + c_{22} + c_{31} + c_{41} + c_{52} + c_{61} + c_{72} + c_{82} + c_{91} = 2930$, $U = v_1 + v_2 + v_3 + v_4 + v_5 + v_6 + v_7 + v_8 + v_9 = 3300$, $p_1 = \theta \times (v_1 + v_2 + v_3) = 584$, $p_2 = \theta \times (v_4 + v_5 + v_6) = 1360$, $p_3 = U - (p_1 + p_2) = 1356$. On the basis of the obtained FC and FC_{client}, we can calculate λ and *Diff* as follows: $\lambda = FC_{client}/FC = 0.80$, $Diff = |\lambda - RBP| = 0.00$, which indicate that under the desirable solution, the distribution proportion of the project financing cost can match the relative bargaining power of the two parties exactly. Besides the results aforementioned, we may calculate NPV_{cont} and NPV_{client} further:

$$\begin{aligned} NPV_{cont} &= \big[p_1 \times \exp(-\alpha \times 6) + p_2 \times \exp(-\alpha \times 11) + p_3 \times \exp(-\alpha \times 14)\big] \\ &\quad - \big[c_{11} \times \exp(-\alpha \times 0) + c_{22} \times \exp(-\alpha \times 4) + c_{31} \times \exp(-\alpha \times 6) \\ &\quad + c_{41} \times \exp(-\alpha \times 10) + c_{52} \times \exp(-\alpha \times 11) + c_{61} \times \exp(-\alpha \times 11) \\ &\quad + c_{72} \times \exp(-\alpha \times 14) + c_{82} \times \exp(-\alpha \times 14) + c_{91} \times \exp(-\alpha \times 14)\big] \\ &= 197.26, \\ NPV_{client} &= NPV_{joint} - NPV_{cont} = 1585.79. \end{aligned}$$

- The non-cooperative case

In this case, it is assumed that the two parties do not know their relative bargaining power very well and make decisions from their own perspectives. In practice, the modes and the completion times of activities are often arranged by the contractor while the payment activities are mainly determined by the client. Therefore, in such a case, the activities will be performed with their cheaper modes by the contractor so that his/her expense, namely TC, can be minimized. In contrast, from the angle of the client, the payment activities will be arranged in the way by which FC_{client} can be reduced as much as possible. This may lead to another feasible solution for the problem: $\Omega = (1, 1, 1, 1, 1, 1, 1, 2, 1)$, $\Gamma = (0, 5, 8, 11, 13, 13, 16, 16, 16)$, and $\Psi = (0, 1, 1, 0, 0, 0, 0, 0, 1)$. In the light of the same process as that applied in the cooperative case, we can obtain NPV_{joint}, FC, FC_{client}, FC_{cont}, TC, λ, *Diff*, NPV_{cont}, and NPV_{client} under the new solution as follows: $NPV_{joint} = 1732.21$, $FC = 512.71$, $FC_{client} = 266.59$, $FC_{cont} = 246.12$, $TC = 2830$, $\lambda = 0.52$, $Diff = 0.28$, $NPV_{cont} = 179.07$, $NPV_{client} = 1553.14$.

Compared with the results in the cooperative case, it can be considered that the contractor and the client in the non-cooperative case have achieved their respective goals to some extent since TC and FC_{client} are both reduced. However, it is unfortunate that their decisions aforementioned decrease rather than increase their profits, i.e. NPV_{cont} and NPV_{client}. The decline of NPV_{cont} derives from the increase of FC_{cont} which is caused by the client's changing the payment activities whereas the decrease in NPV_{client} comes from the postponement of the occurrence of *ER*, which is the result of the contractor's adjusting the modes and the completion times of the activities. The above facts indicate that the contractor and the client can obtain some benefits from their respective decisions, but these decisions may generate greater damages to the profits of the other sides, thus decreasing their NPVs as a whole.

Essentially speaking, the reason for the decreases in the NPVs of the two sides lies in the decline of NPV_{joint}, which is the common source of NPV_{cont} and NPV_{client}. In the cooperative case, NPV_{joint} is maximized and shared by the contractor and the client reasonably according to their relative bargaining power. However, in the non-cooperative case, the objective of maximizing NPV_{joint} is ignored and the two parties pursue to enhance their own profits respectively, without the consideration of the influences on the profit of the other side. Therefore, the results in the cooperative case can be regarded as the global optimization outcomes while those in the non-cooperative case are only the local optimization ones in a sense. On the basis of the above analysis, we can conclude that if the contractor and the client know each other pretty well, in order to get more profits from the project, they will tend to take a cooperative attitude to schedule payments in the project.

3 Simulated annealing algorithm

The model constructed in Sect. 2.1 has two features that render optimal solutions a computationally difficult proposition, especially for real world projects. The model is combinatorial in nature because of the use of binary variables. Furthermore, the number of constraints severely limits the ability to solve the problem to optimality for projects of large size or long duration. For the above reasons, we use a well-known local search metaheuristic, i.e. simulated annealing which is introduced by Metropolis et al. (1953) and is applied to combinatorial optimization for the first time independently by Kirkpatrick et al. (1983) and Cerny (1985), to solve the problem. We choose simulated annealing for two reasons. Firstly, this technique has been successfully applied to a number of project scheduling problems by researchers (e.g. Boctor 1996; Shtub et al. 1996; Etgar et al. 1997; Cho and Kim 1997; Hapke et al. 1998; Viana and Sousa 2000; Dayanand and Padman 2001a; Jozefowska et al. 2001; Bouleimen and Lecocq 2003; Mika et al. 2005; Kavlak et al. 2009; He et al. 2009); and second, it is often characterized by fast convergence and ease of implementation.

Because the problem consists of two subproblems, we divide the algorithm into two modules, namely modules 1 and 2 which are designed to solve subproblems 1 and 2 respectively. In module 1, the desirable Ω and Γ are searched out while in module 2, the desirable Ψ is found on the basis of the desirable Ω and Γ obtained by module 1. In the following of this section, we present the solution representation, the objective function, the preprocessing, the initial solution generation, the neighborhood generation mechanism, and the cooling scheme at first and then describe the design of the two modules.

3.1 Solution representation

Referring to the literatures (Kolisch and Hartmann 1999; Bouleimen and Lecocq 2003; Mika et al. 2008), we represent a solution for the problem, i.e. Ω, Γ, and Ψ, using three lists defined below.

- Execution mode list (EML): This list is composed of n elements and the i-th ($i = 1, 2, \ldots, n$) element defines the mode of activity i.
- Time deviation list (TDL): This list is an n-element list and the i-th ($i = 1, 2, \ldots, n$) element, which is denoted as Δ_i ($\Delta_i \in [0, LF_i - EF_i]$), indicates how many units of activity i's completion time deviating from its possible earliest finish time.
- Payment activity list (PAL): This list includes n 0–1 elements and the i-th ($i = 1, 2, \ldots, n$) element is set at 1 if a payment is attached to activity i and 0 otherwise.

Based on the above three lists, Ω and Ψ are determined by EML and PAL directly while Γ is calculated using the following decoding procedure, where ACT_i denotes activity i's completion time and EAS the set of eligible activities, i.e. the unscheduled activities whose predecessors have been scheduled.

Step 1. Set $EAS = \{activity\ 1\}$, $ACT_1 = 0$.
Step 2. Update EAS, i.e. remove the scheduled activities from EAS and add the new eligible activities into it. If $EAS = \varnothing$ go to step 4; otherwise, go to step 3.
Step 3. On the basis of EML, calculate EF_i for the new eligible activities in EAS and in terms of TDL, determine the completion times of the activities according to the formula $ACT_i = EF_i + \Delta_i$, then go to step 2.
Step 4. Stop the procedure and output all ACT_is, thus obtaining Γ.

3.2 Objective function

Given a solution represented by the above three lists, it is possible that the solution is an infeasible solution where the project deadline constraint is violated. In the algorithms, we transform the deadline constraint into a soft constraint based on a deadline feasibility test function defined as

$$DFT = \max\left\{0, \sum_{t=EF_n}^{LF_n} (y_{nt} t) - D\right\}. \tag{15}$$

During the searching process, if the DFT of a solution is greater than 0 the objective function value of submodel 1 will be penalized according to the following formula:

$$NPV_{joint} = ER\bigl\{\exp\bigl[-\alpha(D + DFT)\bigr]\bigr\} - TEP. \tag{16}$$

Where, TEP is the total expense of the project in which all activities are performed with their most expensive modes.

3.3 Preprocessing

Similar with Sprecher et al. (1997), we adapt the project data to the implementation of simulated annealing using a preprocessing procedure by which all inefficient modes and infeasible modes are eliminated. An inefficient mode is a mode with duration not shorter and, simultaneously, cost not less than any other modes of the considered activity. An infeasible mode is defined as follows. Suppose that EML_{min}^{im} is an EML in which activity i is performed with mode m while other activities are performed with their modes corresponding to the minimal duration. Denote the EF_n under EML_{min}^{im} as $EF_n(\mathrm{EML}_{min}^{im})$. If $EF_n(\mathrm{EML}_{min}^{im}) > D$ mode m is an infeasible mode for activity i, since the project deadline constraint cannot be satisfied no matter how to adjust the modes of other activities.

3.4 Initial solution

After the implementation of the preprocessing procedure, a feasible initial solution can be generated conveniently according to the following steps.

Step 1. Perform each activity with its mode corresponding to the minimal duration, thus obtaining an EML.
Step 2. On TDL, set the values of all the elements at 0.
Step 3. Select randomly $K-1$ activities from all activities except for activities 1 and n. On PAL, set the values of the elements corresponding to the selected activities and activity n at 1 while those of the others at 0.

3.5 Neighborhood generation mechanism

There are three operators utilized in the neighborhood generation mechanism.

- Mode change (MC): On EML, select one activity randomly and change its mode to another executable one for this activity arbitrarily, thus obtaining a neighbor solution. Compute EF_n under the generated solution and judge whether EF_n exceeds the project deadline or not. If the answer is false accept the neighbor solution obtained, otherwise, refuse it and repeat the above operations until the answer becomes false.
- Deviation change (DC): On TDL, choose an activity from all the activities except for activity 1 randomly. Denote the chosen activity as activity i and change the value of $\varDelta_i$ to another possible one within $[0, LF_i - EF_i]$ arbitrarily, thus getting a neighbor solution.
- Element swap (ES): On PAL, except for the last element, select randomly an element from the elements whose values are 1 and swap its position with that of another element chosen randomly from those whose values are 0, changing the current solution to its neighbor.

Note that operators MC and DC are used in module 1 while operator ES is applied in module 2.

3.6 Cooling scheme

The simulated annealing algorithm is specified by the cooling scheme, which is generally constituted of the initial temperature, the cooling function, the Markov chain length, and the stop criterion to terminate the algorithm. A great variety of cooling schemes have now been suggested by different authors and these have been classified in Collins et al. (1988), Eglese (1990), and Koulamas et al. (1994). On the basis of the literatures (Jozefowska et al. 2001; Mika et al. 2005; Kavlak et al. 2009) and a few preliminary computational experiments conducted, the cooling scheme and its parameters used in this study are designed as follows.

- Initial temperature: In module 1, the initial temperature, $Temp^{init}$, is calculated by

$$Temp^{init} = \left(NPV_{joint}^{min} - NPV_{joint}^{init}\right) / \ln Prob^{init}$$

while in module 2 it is computed from

$$Temp^{init} = \left(Diff^{init} - Diff^{max}\right) / \ln Prob^{init}.$$

Where, NPV_{joint}^{init} and $Diff^{init}$ are the objective function values under the initial solutions for submodels 1 and 2 respectively, NPV_{joint}^{min} and $Diff^{max}$ are the minimal objective function value of submodel 1 and the maximal objective function value of submodel 2 among the 50 neighbor solutions of the initial solution respectively, and $Prob^{init}$, which is set at 0.95 in this application, is the initial acceptance ratio defined as the number of accepted neighbors divided by that of the proposed neighbors.

- Cooling rate: Beginning from the initial value, the temperature is progressively reduced according to a given cooling rate, which is set at 0.9 in our implementation.
- Markov chain length: The lengths of Markov chains, which determine the number of transitions for a given value of the temperature, are set at $12 \cdot n$ and $10 \cdot n$ in modules 1 and 2 respectively.
- Stop criterion: The search process terminates when the current temperature drops to the final temperature, which is set at 0.01 in the two modules.

3.7 Modules 1 and 2

In the light of different searching structures used, we design two versions of module 1, which are module 1 with random sampling (M1RS) and module 1 with nested loops (M1NL) respectively. The so-called M1RS means that in the searching process, operators MC and DC are selected randomly and applied independently to the current solution to generate a neighbor solution for subproblem 1. However, in M1NL, the above operators are utilized in a nested way, based on the fact that activities' completion times can not be arranged until their modes are all determined while the reverse is not true. More concretely, M1NL is composed of the following two nested loops.

- INNERLOOP(EML): In this loop, the EML is given and the task is to search the desirable TDL.
- OUTERLOOP: With the results obtained by INNERLOOP(EML), this loop seeks the desirable EML, forming a desirable solution for subproblem 1.

On the basis of the desirable EML and TDL found by module 1, module 2 searches the desirable PAL for subproblem 2, thus constituting together a desirable solution for the problem.

4 Computational experiment

4.1 Experimental design

For the sake of description, we represent the simulated annealing algorithm with M1RS and M1NL as SA-M1RS and SA-M1NL respectively in the following of the paper. In the experiment, SA-M1NL utilizes the stop criterion defined in Sect. 3.6, and during its searching process, the total number of the feasible solutions visited by modules 1 and 2 are recorded and denoted as Num^1_{stop} and Num^2_{stop} respectively. To compare the performances of SA-M1RS and SA-M1NL on a common basis, we adjust the stop criterion of SA-M1RS as follows: The algorithm stops and outputs the current solution as the desirable one when the number of the feasible solutions visited by modules 1 and 2 reaches Num^1_{stop} and Num^2_{stop} respectively. Note that in module 1 of SA-M1RS, since the two decision vectors are searched concurrently the length of Markov chains, L, is set at $140 \cdot n$.

To evaluate the performances of SA-M1RS and SA-M1NL, we utilize a simple heuristic algorithm, namely the multi-start iterative improvement (MSII) (Mika et al. 2008; Waligóra 2008, 2009), to generate benchmark schedules for comparison. For the two subproblems, MSII starts from the same initial solution and employs the same neighbor generation mechanism as those used in SA-M1RS. In the searching course of MSII, the most improving neighbor solution is chosen and when there are no improving moves, it restarts with another feasible solution generated randomly. The algorithm terminates and takes the best solution

Table 1 Parameter setting used to generate the data set

Parameter	Setting
Number of non-dummy activities	10, 20, 30, or 40
Number of instances generated under a given number of non-dummy activities	20
Number of initial and terminal activities	Randomly selected from 2, 3, and 4
Maximal number of successors and predecessors	5
Number of performing modes	3
Activity durations under mode 1, d_{i1}	Randomly selected from interval [1, 10]
Activity costs under mode 1, c_{i1}	Randomly selected from interval [10, 20]
Activity durations under mode 2, d_{i2}	$\rho_1 \cdot d_{i1}$, where ρ_1 is randomly selected from interval [0.8, 1]
Activity costs under mode 2, c_{i2}	$\rho_2 \cdot c_{i1}$, where ρ_2 is randomly selected from interval [1, 1.2]
Activity durations under mode 3, d_{i3}	$\rho_3 \cdot d_{i2}$, where ρ_3 is randomly selected from interval [0.8, 1]
Activity costs under mode 3, c_{i3}	$\rho_4 \cdot c_{i2}$, where ρ_4 is randomly selected from interval [1, 1.2]
Earned values of activities, v_i	$\rho_5 \cdot c_{i3}$, where ρ_5 is randomly selected from interval [1.3, 1.5]
Client's expected revenue, *ER*	$\rho_{ER} \cdot U$, where ρ_{ER} is randomly selected from 1.6, 1.8, and 2.0
Relative bargaining power, *RBP*	Randomly selected from 0.4, 0.5, and 0.6

Table 2 Levels of the key parameters

Parameter	Value
Interest rate per period, α	0.01, 0.02, and 0.03
Project deadline, D	$\rho_D \cdot (CL_{max} - CL_{min}) + CL_{min}$, where CL_{max} and CL_{min} are the length of the critical path of the network when all the activities are performed with modes 1 and 3 respectively and ρ_D is set at 0.4, 0.6, and 0.8
Financing cost rate, β	0.01, 0.02, and 0.03
Payment number, K	3, 4, and 5
Compensation proportion, θ	0.7, 0.8, and 0.9

found as the desirable one when the numbers of the visited feasible solutions for subproblems 1 and 2 reach Num^1_{stop} and Num^2_{stop}, respectively.

The three algorithms are tested on a data set constructed by ProGen project generator (Kolisch et al. 1995; Kolisch and Sprecher 1996). The set consists of 80 instances and the parameter setting used to generate the instances is described in Table 1. The value of the key parameters, including α, D, β, K, and θ, is set at three levels given in Table 2. A full factorial experiment of the five parameters with three levels results in 3^5 replicates for each instance and $80 \times 3^5 = 19{,}440$ ones overall. Note that during the generation of the instances, it is ensured that there are no inefficient modes for activities through the parameter setting. However, infeasible modes, which have to be eliminated by the preprocessing procedure presented in Sect. 3.3, cannot be avoided completely since for a certain instance the parameter D needs to be set at different values. The following five indices are defined to evaluate the performances of the algorithms.

- *NUM*: The number of instances for which the algorithm find a solution equal to the best solution known, i.e. the best solution found by any of the algorithms.
- *ARP*: Average relative percent below the known best solution for the problems.
- *MRP*: Maximal relative percent below the known best solution for the problems.
- *ACT*: Average computational time of the algorithm for solving the problems.
- *MCT*: Maximal computational time of the algorithm for solving the problems.

Note that since the FCDPPSP consists of two subproblems we may obtain two groups of the above indices in the experiment. The algorithms are coded and compiled with Visual Basic 6.0 and the computational experiment is performed on a Pentium-based personal computer with 1.60 GHz clock-pulse and 256 MB RAM.

4.2 Experimental results

The computational results are presented in Table 3. For subproblem 1, Table 3 makes clear that *ARP*s and *MRP*s for SA-M1RS and SA-M1NL are much less whereas *NUM*s for the two algorithms are much greater than the corresponding indices for MSII, in particular when the instances become larger. This indicates that SA-M1RS and SA-M1NL outperform MSII remarkably and their superiority augments with the increase of the problem scale. This result is not surprising because the intelligent search procedures generally get an advantage over the simple search method, and this advantage grows as the problem becomes more complex. As for the two simulated annealing algorithms, it can be seen that the three indices aforementioned for SA-M1NL are a bit better than those for SA-M1RS, and with the increase of the number of activities, the indices for SA-M1NL improve while those for SA-M1RS deteriorate. So, we can say that compared with SA-M1RS, SA-M1NL performs a little better in general, especially for the larger instances. The above outcomes may be explained as follows: SA-M1NL is based on the characteristics of the problem and utilizes the two nested loops to search the desirable solutions while in SA-M1RS, the two neighbor generation operators are selected in a random fashion. Therefore, it can be considered that the searching structure of SA-M1NL may be more reasonable than that of SA-M1RS, making it own a higher searching efficiency than SA-M1RS. The results for subproblem 2, which is much easier to be solved than subproblem 1, are relatively simple. Since SA-M1RS and SA-M1NL employ the same module 2, their *NUM*s, *ARP*s, and *MRP*s for subproblem 2 are almost identical. Compared with SA-M1RS and SA-M1NL, MSII gets a bit worse outcome because its searching process is less intelligent than those of SA-M1RS and SA-M1NL.

The computational times are ones to be expected—MSII runs faster than SA-M1RS and SA-M1NL since it owns a simpler searching process than the other two algorithms. Concerning SA-M1RS and SA-M1NL, because the searching structure of the latter is more complex than that of the former it seems a little strange that the former works longer than the latter for tackling subproblem 1. Recall that during the searching process of the algorithms, the two neighbor generation operators, i.e. MC and DC, are utilized to generate neighbor solutions based on the current one. In SA-M1RS, when operator MC is selected to generate neighbor solutions, the EML is changed while the TDL remains unchanged. Because the value of $LF_i - EF_i$ is determined by the EML, this may lead to the result that $\Delta_i > LF_i - EF_i$, causing the generated solution to be infeasible. However, in SA-M1NL, this shortcoming is eliminated by the loop nested searching structure where operator DC works under a given EML. The facts aforementioned may explain why SA-M1NL runs faster than SA-M1RS to some extent although the former owns a more complex searching structure than the latter. Summarizing the computational results discussed above, we can draw the conclusion that

Table 3 Computational results of the experiment

	n	SA-M1RS					SA-M1NL					MSII				
		NUM	*ARP* (%)	*MRP* (%)	*ACT* (s)	*MCT* (s)	*NUM*	*ARP* (%)	*MRP* (%)	*ACT* (s)	*MCT* (s)	*NUM*	*ARP* (%)	*MRP* (%)	*ACT* (s)	*MCT* (s)
Subproblem 1	12	4688	0.08	0.24	1.12	1.69	4678	0.09	0.19	0.94	1.47	4366	0.20	0.48	0.54	1.11
	22	4269	0.93	2.11	7.57	10.83	4313	0.78	1.73	6.51	10.19	3265	5.26	10.36	4.31	6.88
	32	4005	1.65	4.05	37.54	56.66	4418	0.47	1.11	35.95	54.00	1896	10.34	17.54	22.26	35.64
	42	3671	3.11	7.73	104.22	168.94	4676	0.05	0.11	100.64	151.23	982	22.61	36.44	64.44	96.62
Subproblem 2	12	4860	0.00	0.00	0.07	0.15	4860	0.00	0.00	0.08	0.14	4846	0.00	0.00	0.06	0.12
	22	4860	0.00	0.00	0.25	0.43	4860	0.00	0.00	0.26	0.47	4758	0.00	0.02	0.20	0.41
	32	4860	0.00	0.00	0.76	1.27	4860	0.00	0.00	0.73	1.28	4621	0.02	0.05	0.53	1.11
	42	4860	0.00	0.00	1.32	2.51	4860	0.00	0.00	1.33	2.48	4536	0.04	0.11	1.03	2.07

Table 4 Influences of the key parameters on the computational results

Parameter	Value	SA-M1RS						SA-M1NL						MSII					
		NPV_{joint}	NPV_{cont}	NPV_{client}	*Diff*	FC_{cont}	FC_{client}	NPV_{joint}	NPV_{cont}	NPV_{client}	*Diff*	FC_{cont}	FC_{client}	NPV_{joint}	NPV_{cont}	NPV_{client}	*Diff*	FC_{cont}	FC_{client}
α	0.01	364.55	75.17	289.38	0.025	41.93	47.31	367.73	76.03	291.90	0.023	41.23	45.66	333.33	68.24	265.10	0.024	42.83	48.28
	0.02	295.28	57.57	237.71	0.022	42.61	45.20	299.39	58.21	240.95	0.023	41.75	43.82	270.46	53.77	216.61	0.026	41.42	46.54
	0.03	165.63	34.43	131.20	0.023	41.60	46.55	167.46	34.74	132.72	0.024	41.28	46.76	144.29	29.36	114.75	0.024	43.64	49.20
ρ_D	0.4	273.21	51.55	221.97	0.025	43.47	46.40	275.28	51.68	223.76	0.024	43.45	45.48	248.37	45.33	202.82	0.023	43.59	47.83
	0.6	275.47	56.59	218.87	0.022	41.82	45.82	279.30	58.11	221.40	0.021	40.91	44.21	249.46	52.55	197.03	0.027	42.51	48.13
	0.8	276.52	59.00	217.51	0.022	41.21	46.67	279.88	59.21	220.43	0.022	39.34	46.93	250.15	53.40	196.51	0.022	41.90	48.28
β	0.01	275.22	55.06	220.13	0.024	28.53	32.05	277.72	56.18	221.75	0.020	29.66	33.42	249.37	50.24	199.05	0.025	31.34	34.49
	0.02	274.68	55.63	218.95	0.023	39.09	44.37	277.89	56.70	220.99	0.025	37.78	42.24	248.56	50.33	198.56	0.023	38.17	46.41
	0.03	275.50	56.21	219.37	0.023	58.38	62.49	278.67	56.32	222.67	0.023	57.24	60.62	249.74	50.57	198.85	0.023	58.45	63.09
K	3	275.14	53.29	222.05	0.032	43.12	45.74	278.49	54.55	223.74	0.030	42.27	44.31	249.23	48.58	200.35	0.031	43.27	47.11
	4	275.29	55.56	219.41	0.023	41.73	46.20	277.77	56.48	221.30	0.021	41.34	45.80	249.76	50.77	198.94	0.025	42.84	48.21
	5	274.76	58.09	216.96	0.014	41.31	46.66	278.38	58.21	220.13	0.016	40.65	46.63	249.19	52.00	197.17	0.016	41.63	48.79
θ	0.7	274.54	51.10	223.68	0.023	44.33	44.51	278.12	51.13	226.86	0.025	43.25	42.66	248.67	46.41	202.33	0.023	44.31	45.11
	0.8	275.29	55.78	219.70	0.022	42.12	46.72	278.35	56.63	221.92	0.020	41.61	46.02	249.44	50.39	199.06	0.025	42.78	48.09
	0.9	275.16	60.49	214.97	0.024	39.76	47.93	278.00	61.31	216.80	0.024	39.48	47.72	249.63	54.67	194.97	0.024	40.88	50.84

SA-M1NL is the most promising procedure for the problem among the three algorithms presented.

The influences of the key parameters on average NPV_{joint}, NPV_{cont}, NPV_{client}, *Diff*, FC_{cont}, and FC_{client} are presented in Table 4, from which the following phenomena can be observed. Firstly, with the growth of α, NPV_{joint}, NPV_{cont}, and NPV_{client} decrease. This is an immediate observation when formula (3) is taken into account. Secondly, as ρ_D increases, NPV_{joint} and NPV_{cont} climb while NPV_{client} and FC_{cont} drop. The reasons for these results are as follows. When ρ_D increases D augments subsequently, so, the project deadline constraint is loosened to some degree, causing NPV_{joint} improved. In addition, the postponement of D can make the contractor execute more activities with their cheaper modes, therefore, NPV_{cont} rises and FC_{cont} falls. However, for the client, because the occurrence of his/her expected revenue, which is assumed to take place at the completion of the project, may be delayed in such a case, NPV_{client} declines. Thirdly, increasing β makes FC_{cont} and FC_{client} increased monotonously and this can be explained by formulae (1) and (2) directly. Fourthly, when K increases NPV_{cont} and FC_{client} ascend whereas NPV_{client}, FC_{cont}, and *Diff* descend. This is because that the increase of K can make the payments occur at earlier activities and thus plays a negative effect on the client whereas a positive effect on the contractor, resulting in NPV_{cont} and FC_{client} increased while NPV_{client} and FC_{cont} decreased. Furthermore, increasing K may generate more choices on the arrangements of payments at activities, therefore, the obtained *Diff* cannot be worse than the one obtained for the smaller K in general. At last, as θ increases, NPV_{cont} and FC_{client} go up while NPV_{client} and FC_{cont} go down. This is a reasonable result since enhancing the value of θ leads to greater parts of payments to occur earlier during the execution of the project.

5 Conclusions

This paper studies the project payment scheduling problem based on the distribution of the project financing cost which is defined as the expense for raising money from the outside or the opportunity cost of the capital devoted into the project. The purpose is to find the project payment schedule that can not only maximize the joint revenue of the contractor and the client but also match their relative bargaining power and thus be accepted by them. Based on its identification, we divide the studied problem into two subproblems, i.e. how to determine the modes of activities and the completion times of activities so that the joint revenue is maximized and how to arrange payments at activities so as to distribute the project financing cost according to their relative bargaining power. An optimization model, which consists of two corresponding submodels, is constructed using the activity-based method and an example is utilized to illustrate the practical utility of the constructed model. For the strong NP-hardness of the problem two simulated annealing heuristic algorithms, namely SA-M1RS and SA-M1NL which own different searching structures, are developed and compared with MSII on the basis of a computational experiment performed on a data set generated by ProGen. From the computational results, the following conclusion is drawn: The proposed SA-M1NL is the most promising algorithm for solving the defined problem, surpassing SA-M1RS somewhat while clearly outperforming MSII especially when the instances become larger.

In addition, the influences of some key parameters on the computational results are investigated through the full factorial experiment of the five parameters with three levels, and the obtained observations are described below: Increasing the interest rate decreases the contractor's, the client's, and their joint NPVs; When the project deadline is delayed the contractor's and the joint NPVs climb while client's NPV and the contractor's financing cost drop; Increasing financing cost rate increases the contractor's and the client's financing costs; With the growth of the payment number or the compensation proportion, the contractor's NPV and the client's financing cost ascend whereas the client's NPV and the contractor's financing cost descend; The absolute difference between the distribution proportion of the project financing cost and the relative bargaining power of the two parties tends to go down as the payment number goes up.

Note that the research in this paper is based on the joint perspective of the two parties. Although it is not always the case that the contractor and the client take a cooperative attitude in the real-world project management, the research can still provide some supports for the coordination of the conflict between the two parties in contract negotiation and facilitate them to achieve a more profitable outcome.

Acknowledgements The research was funded by the National Natural Science Foundation of China under contract No. 70971105. We express our gratitude to two anonymous referees for their valuable comments, which have improved this paper considerably.

References

Błażewicz, J., Ecker, K., Pesch, E., Schmidt, G., & Węglarz, J. (2007). *Handbook on scheduling: from theory to applications*. Berlin: Springer.

Boctor, F. F. (1996). Resource-constrained project scheduling by simulated annealing. *International Journal of Production Research*, *34*(8), 2335–2351.

Bouleimen, K., & Lecocq, H. (2003). A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version. *European Journal of Operational Research*, *149*(2), 268–281.

Brucker, P., Drexl, A., Möhring, R., Neumann, K., & Pesch, E. (1999). Resource-constrained project scheduling: notation, classification, models and methods. *European Journal of Operational Research*, *112*(1), 3–41.

Cerny, V. (1985). Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm. *Journal of Optimization Theory and Applications*, *45*(1), 41–51.

Cho, J. H., & Kim, Y. D. (1997). A simulated annealing algorithm for resource constrained project scheduling problems. *Journal of the Operational Research Society*, *48*(7), 736–744.

Collins, N. E., Eglese, R. W., & Golden, B. L. (1988). Simulated annealing—an annotated bibliography. *American Journal of Mathematical and Management Sciences*, *8*(3–4), 209–307.

Dayanand, N., & Padman, R. (1997). On modeling progress payments in project networks. *Journal of the Operational Research Society*, *48*(9), 906–918.

Dayanand, N., & Padman, R. (2001a). A two stage search heuristic for scheduling payments in projects. *Annals of Operations Research*, *102*(1), 197–220.

Dayanand, N., & Padman, R. (2001b). Project contracts and payments schedules: the client's problem. *Management Science*, *47*(12), 1654–1667.

Demeulemeester, E., & Herroelen, W. (2002). *Project scheduling—a research handbook*. Boston: Kluwer Academic.

De, P., Dunne, E. J., Ghosh, J. B., & Wells, C. E. (1995). The discrete time-cost tradeoff problem revisited. *European Journal of Operational Research*, *81*(2), 225–238.

De, P., Dunne, E. J., Ghosh, J. B., & Wells, C. E. (1997). Complexity of the discrete time-cost tradeoff problem for project networks. *Operations Research*, *45*(2), 302–306.

Drezet, L.-E. (2008). RCPSP with financial costs. In C. Artigues, S. Demassey, & E. Néron (Eds.), *Resource-constrained project scheduling: models, algorithms, extensions and applications* London: ISTE-Wiley.

Eglese, R. W. (1990). Simulated annealing: a tool for operational research. *European Journal of Operational Research*, *46*(3), 271–281.

Etgar, R., Shtub, A., & LeBlanc, L. J. (1997). Scheduling projects to maximize net present value—the case of time-dependent, contingent cash flows. *European Journal of Operational Research*, *96*(1), 90–96.

Hapke, M., Jaszkiewicz, A., & Słowiński, R. (1998). Interactive analysis of multiple-criteria project scheduling problems. *European Journal of Operational Research*, *107*(2), 315–324.

Hartmann, S., & Briskorn, D. (2010). A survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research*, *207*(1), 1–14.

Herroelen, W. S., Dommelen, P., & Demeulemeester, E. L. (1997). Project network models with discounted cash flows: a guided tour through recent developments. *European Journal of Operational Research*, *100*(1), 97–121.

He, Z., & Xu, Y. (2008). Multi-mode project payment scheduling problems with bonus-penalty structure. *European Journal of Operational Research*, *189*(3), 1191–1207.

He, Z., Wang, N., Jia, T., & Xu, Y. (2009). Simulated annealing and tabu search for multi-mode project payment scheduling. *European Journal of Operational Research*, *198*(3), 688–696.

Icmeli, O., Erenguc, S. S., & Zappe, C. J. (1993). Project scheduling problem: a survey. *Journal of Operations and Productions Management*, *13*(11), 80–91.

Jozefowska, J., Mika, M., Rozycki, R., Waligora, G., & Weglarz, J. (2001). Simulated annealing for multi-mode resource-constrained project scheduling. *Annals of Operations Research*, *102*(1), 137–155.

Kavlak, N., Ulusoy, G., Şerifoğlu, F. S., & Birbil, Ş. İ. (2009). Client-contractor bargaining on net present value in project scheduling with limited resources. *Naval Research Logistics*, *56*(2), 93–112.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, *220*(4598), 671–680.

Kolisch, R., & Hartmann, S. (1999). Heuristic algorithms for the resource-constrained project scheduling problem: classification and computational analysis. In J. Węglarz (Ed.), *Project scheduling: recent models, algorithms, and applications* Boston: Kluwer Academic.

Kolisch, R., & Padman, R. (2001). An integrated survey of deterministic project scheduling. *Omega*, *29*(3), 249–272.

Kolisch, R., & Sprecher, A. (1996). PSPLIB—a project scheduling problem library. *European Journal of Operational Research*, *96*(1), 205–216.

Kolisch, R., Sprecher, A., & Drexl, A. (1995). Characterization and generation of a general class of resource-constrained project scheduling problems. *Management Science*, *41*(10), 1693–1703.

Koulamas, C., Antony, S. R., & Jean, R. (1994). A survey of simulated annealing applications to operations research problems. *Omega*, *22*(1), 44–56.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., & Teller, E. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, *21*(6), 1087–1092.

Mika, M., Waligóra, G., & Węglarz, J. (2005). Simulated annealing and tabu search for multi-mode resource-constrained project scheduling with positive discounted cash flows and different payment models. *European Journal of Operational Research*, *164*(3), 639–668.

Mika, M., Waligóra, G., & Węglarz, J. (2008). Tabu search for multi-mode resource-constrained project scheduling with schedule-dependent setup times. *European Journal of Operational Research*, *187*(3), 1238–1250.

Özdamar, L., & Ulusoy, G. (1995). A survey on the resource-constrained project scheduling problem. *IIE Transactions*, *27*(5), 574–586.

Russell, A. H. (1970). Cash flows in networks. *Management Science*, *16*(5), 357–373.

Shtub, A., LeBlanc, L. J., & Cai, Z. (1996). Scheduling programs with repetitive projects: a comparison of a simulated annealing, a genetic and a pair-wise swap algorithm. *European Journal of Operational Research*, *88*(1), 124–138.

Sprecher, A., Hartmann, S., & Drexl, A. (1997). An exact algorithm for project scheduling with multiple modes. *OR-Spektrum*, *19*(3), 195–203.

Szmerekovsky, J. G. (2005). The impact of contractor behavior on the client's payment-scheduling problem. *Management Science*, *51*(4), 629–640.

Ulusoy, G., & Cebelli, S. (2000). An equitable approach to the payment scheduling problem in project management. *European Journal of Operational Research*, *127*(2), 262–278.

Viana, A., & Sousa, J. P. (2000). Using metaheuristics in multiobjective resource constrained project scheduling. *European Journal of Operational Research*, *120*(2), 359–374.

Waligóra, G. (2008). Discrete-continuous project scheduling with discounted cash flows—a tabu search approach. *Computers & Operations Research*, *35*(7), 2141–2153.

Waligóra, G. (2009). Tabu search for discrete-continuous scheduling problems with heuristic continuous resource allocation. *European Journal of Operational Research*, *193*(3), 849–856.

Węglarz, J., Józefowska, J., Mika, M., & Waligóra, G. (2011). Project scheduling with finite or infinite number of activity processing modes—a survey. *European Journal of Operational Research*, *208*(3), 177–205.